



**TUGAS AKHIR - KI141502**

**RANCANG BANGUN APLIKASI SIMULASI JARINGAN  
UNTUK PRAKTIKUM JARINGAN KOMPUTER  
MEMANFAATKAN SCAPY DALAM LINGKUNGAN LXC**

**DHIMAS BAGUS PRAMUDYA  
NRP 5111100044**

**Dosen Pembimbing I  
Royyana Muslim Ijtihadie, S.Kom, M.Kom, PhD.**

**Dosen Pembimbing II  
Baskoro Adi Pratomo, S.Kom, M.Kom**

**JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015**



**UNDERGRADUATE THESES - KI141502**

# **DESIGN AND IMPLEMENTATION OF NETWORK SIMULATOR FOR COMPUTER NETWORK LABWORK WITH SCAPY IN LXC ENVIRONMENT**

**DHIMAS BAGUS PRAMUDYA  
NRP 5111100044**

**Supervisor I  
Royyana Muslim Ijtihadie, S.Kom, M.Kom, PhD.**

**Supervisor II  
Baskoro Adi Pratomo, S.Kom, M.Kom.**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2015**

## LEMBAR PENGESAHAN

### RANCANG BANGUN APLIKASI SIMULASI JARINGAN UNTUK PRAKTIKUM JARINGAN KOMPUTER MEMANFAATKAN SCAPY DALAM LINGKUNGAN LXC

#### TUGAS AKHIR

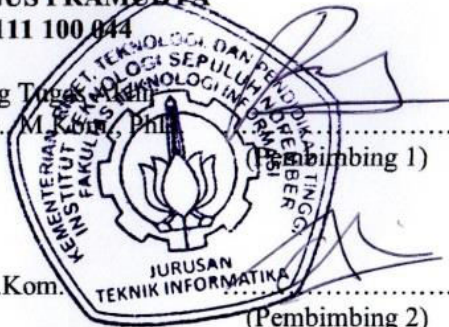
Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Arsitektur dan Jaringan Komputer  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**DHIMAS BAGUS PRAMUDYA**  
**NRP . 5111 100 044**

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D. ....  
NIP. 197708242006041001 (Pembimbing 1)
2. Baskoro Adi Pratomo, S.Kom., M.Kom. ....  
NIP: 198702182014041001 (Pembimbing 2)



**SURABAYA**  
**Desember, 2015**

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“RANCANG BANGUN APLIKASI SIMULASI JARINGAN UNTUK PRAKTIKUM JARINGAN KOMPUTER MEMANFAATKAN SCAPY DALAM LINGKUNGAN LXC”**. Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT atas anugerahnya yang tidak terkira kepada penulis dan Nabi Muhammad SAW.
2. Bapak, Ibu, Galih dan Rizky yang telah memberikan dukungan moral dan material serta doa yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan tugas akhir ini.
3. Bapak Royyana Muslim Ijtihadie, S.Kom, M.Kom, PhD. selaku pembimbing I yang telah membantu, membimbing, dan memotivasi penulis dalam menyelesaikan Tugas Akhir ini dengan sabar.
4. Bapak Baskoro Adi Pratomo, S.Kom, M.Kom. selaku pembimbing II yang juga telah membantu, membimbing, dan memotivasi penulis dalam mengerjakan Tugas Akhir ini.

5. Bapak Darlis Herumurti, S.Kom, M.Komselaku Kepala Jurusan Teknik Informatika ITS, Bapak Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.
6. Teman-teman Laboratorium AJK, Uyung, Harum, Romen, Dimas, Eva, Si Ganteng Sangat, Agus, Thiar, Adhipur, Nisa, Zaza, Wicak, Ahmad, Setyo, Saragih, Nindi, Syukron, Fatih, Risma, dan Fourir yang senantiasa menghibur dan mendukung penulis dalam mengerjakan tugas akhir ini serta menemani penulis di laboratorium.
7. Teman-teman sejawat, Arman, Chibgoh, dan Novalia, yang menemani penulis dikala suka dan duka.
8. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Desember 2015

Dhimas Bagus Pramudya

# **RANCANG BANGUN APLIKASI SIMULASI JARINGAN UNTUK PRAKTIKUM JARINGAN KOMPUTER MEMANFAATKAN SCAPY DALAM LINGKUNGAN LXC**

**Nama Mahasiswa** : DHIMAS BAGUS PRAMUDYA  
**NRP** : 5111100044  
**Jurusan** : Teknik Informatika FTIF-ITS  
**Dosen Pembimbing 1** : Royyana Muslim Ijtihadie, S.Kom,  
M.Kom, PhD.  
**Dosen Pembimbing 2** : Baskoro Adi Pratomo, S.Kom,  
M.Kom.

## **Abstrak**

*Praktikum Jaringan Komputer yang sudah berlangsung di Jurusan Teknik Informatika dalam mensimulasikan topologi jaringan menggunakan User Mode Linux (UML). Namun server mengalami penurunan kinerja, ketika mensimulasikan topologi dalam jumlah banyak yang menggunakan host dan router. Hal tersebut mengakibatkan waktu startup yang lama, latency yang tinggi, dan kecenderungan error sering dialami server .*

*Linux Container merupakan lingkungan virtual berbasis container yang menyediakan ukuran image yang lebih kecil dari User Mode Linux. Proses startup pada Linux Container lebih cepat dibandingkan User Mode Linux meskipun banyak aplikasi server yang sudah dipasang. Dengan kelebihan Linux Container diatas, kecenderungan error yang diakibatkan oleh User Mode Linux dapat dikurangi. Selain itu dalam praktikum jaringan komputer manipulasi paket dengan scapy dapat dimanfaatkan untuk memperdalam praktikan dalam menganalisa paket secara detail apa saja yang dikirim dan respon yang diberikan oleh paket tersebut. Oleh karena itu dalam Tugas Akhir ini diusulkan lingkungan virtualisasi yaitu Linux Container (LXC) dan simulasi pengiriman paket menggunakan Scapy.*

*Dari hasil uji coba dengan sistem simulasi jaringan ini, didapatkan waktu respon untuk menangani setiap rute sekitar kurang dari satu detik sampai tujuh detik. Komputer web server yang digunakan mampu membuat virtualisasi sebanyak 400 dengan web server yang sedang diakses oleh 20 pengguna.*

**Kata kunci : Praktikum, Simulasi Jaringan, Linux Container, Scapy**

# **DESIGN AND IMPLEMENTATION OF NETWORK SIMULATOR FOR COMPUTER NETWORK LABWORK WITH SCAPY IN LXC ENVIRONMENT**

**Student's Name** : DHIMAS BAGUS PRAMUDYA  
**Student's ID** : 5111100044  
**Department** : Teknik Informatika FTIF-ITS  
**First Advisor** : Royyana Muslim Ijtihadie, S.Kom,  
M.Kom, PhD.  
**Second Advisor** : Baskoro Adi Pratomo, S.Kom,  
M.Kom.

## **Abstract**

*Network computing labwork that held in Informatics Department use User Mode Linux for network topology simulation. But performance of server that handle User Mode Linux decreased while serve many virtualization of host and router. Because decreasing performance, startup of virtualization take long time, high latency, and errors often arise from server .*

*Linux container is a virtual environment based container that provide image size smaller than User Mode Linux. Startup process of Linux Container is faster than the User Mode Linux though many application server s that have been installed . With the advantages Linux Container server error from User Mode Linux can be reduced . In addition, packet manipulation in labwork computer network with scapy can be utilized to deepen the practitioner in analyze the network package in detail, what is sent and the response given by the network package. Therefore, in this final project proposed virtualization environment using Linux Containers ( LXC ) and simulation of network package delivery using scapy*

*By the reference of experiment results toward this network simulation system, obtained the response time to handle each*



*route approximately less than five second. Web server computer that provide this simulation can handle 400 virtualization though 20 user access it.*

**Keyword : Labwork, Network Simulation, Linux Container, Scapy**

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak .....	vii
Abstract .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL.....	xvii
DAFTAR KODE SUMBER .....	xix
BAB I PENDAHULUAN .....	21
1.1 Latar Belakang.....	21
1.2 Rumusan Masalah .....	22
1.3 Batasan Masalah.....	22
1.4 Tujuan.....	23
1.5 Manfaat.....	23
1.6 Metodologi .....	23
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	24
BAB II TINJAUAN PUSTAKA.....	27
2.1 Simulasi Jaringan.....	27
2.2 Linux Container (LXC) .....	27
2.3 Scapy .....	29
2.4 Flask .....	30
2.5 GoJS .....	30
2.6 Bridge-utils.....	31
BAB III DESAIN DAN PERANCANGAN .....	33
3.1 Kasus Penggunaan.....	33
3.2 Deskripsi Fitur .....	35
3.2.1 Fitur untuk Praktikan.....	35
3.2.2 Fitur untuk Admin .....	36
3.3 Arsitektur Sistem .....	36
3.3.1 Desain Umum Sistem.....	37
3.3.2 Desain <i>Backend</i> .....	37

3.3.3	Desain <i>Frontend</i> .....	39
3.3.4	Desain Linux Container dan Bridge-utils .....	41
3.3.5	Desain Basis Data .....	42
3.3.6	Desain Manipulasi Paket dengan Scapy .....	43
BAB IV IMPLEMENTASI .....		47
4.1	Lingkungan Implementasi .....	47
4.2	Implementasi <i>Backend</i> .....	47
4.3	Implementasi <i>Frontend</i> .....	53
4.3.1	Antarmuka Praktikan .....	53
4.3.2	Antarmuka Admin .....	55
4.4	Implementasi Linux Container dan Bridge-utils .....	57
4.5	Implementasi Basis Data .....	58
4.6	Implementasi Manipulasi Paket dengan Scapy .....	60
BAB V UJI COBA DAN EVALUASI .....		62
5.1	Lingkungan Uji Coba .....	62
5.2	Skenario Uji Coba .....	63
5.2.1	Uji Fungsionalitas .....	64
5.2.2	Uji Fungsionalitas dengan Skenario Modul Praktikum .....	67
5.2.3	Uji Kapasitas dan Performa .....	67
5.3	Hasil Uji Coba dan Evaluasi .....	70
5.3.1	Hasil Uji Fungsionalitas .....	70
5.3.2	Hasil Uji Fungsionalitas dengan Skenario Modul Praktikum .....	74
5.3.3	Hasil Uji Kapasitas dan Performa .....	76
5.3.4	Evaluasi .....	78
BAB VI KESIMPULAN DAN SARAN .....		80
6. 1.	Kesimpulan .....	80
6. 2.	Saran .....	80
DAFTAR PUSTAKA .....		82
LAMPIRAN .....		84
B.	Hasil Uji Coba .....	85
BIODATA PENULIS .....		86

## DAFTAR GAMBAR

Gambar 3.2.1 Diagram Kasus Penggunaan .....	35
Gambar 3.3.1 Desain Arsitektur Sistem .....	40
Gambar 3.3.2 Peta Situs Halaman Praktikan .....	42
Gambar 3.3.3 Peta Situs Halaman Admin .....	42
Gambar 3.3.4 Desain Komponen Virtualisasi .....	43
Gambar 4.5.1 Implementasi Komponen Virtualisasi .....	59
Gambar 4.6.1 Tampilan <i>Form</i> Paket Protokol ICMP .....	60
Gambar 4.6.2 Tampilan <i>Form</i> Paket Protokol TCP .....	60
Gambar 5.2.1 Desain Arsitektur Uji Fungsionalitas .....	63
Gambar 5.2.2 Skenario Modul <i>Web Server</i> .....	67
Gambar 5.2.3 Desain Arsitektur Uji Kapasitas dan Performa Komputer <i>Web Server</i> .....	68
Gambar A.1 Tangkapan Layar Hasil Uji Coba Komputer <i>Web Server</i> dengan 100 Virtualisasi .....	83
Gambar A.2 Tangkapan Layar Hasil Uji Coba Komputer <i>Web Server</i> dengan 200 Virtualisasi .....	83
Gambar A.3 Tangkapan Layar Hasil Uji Coba Komputer <i>Web Server</i> dengan 300 Virtualisasi .....	83
Gambar A.4 Tangkapan Layar Hasil Uji Coba Komputer <i>Web Server</i> dengan 400 Virtualisasi .....	83
Gambar A.5 Tangkapan Layar Hasil Uji Coba Komputer <i>Web Server</i> dengan 485 Virtualisasi .....	84
Gambar B.1 Tangkapan Layar Hasil Uji Coba Pembanding dengan 10 Virtualisasi .....	84
Gambar B.2 Tangkapan Layar Hasil Uji Coba Pembanding dengan 50 Virtualisasi .....	84
Gambar B.3 Tangkapan Layar Hasil Uji Coba Pembanding dengan 100 Virtualisasi .....	84
Gambar B.4 Tangkapan Layar Hasil Uji Coba Pembanding dengan 150 Virtualisasi .....	84

## DAFTAR TABEL

Tabel 2.2.1 Daftar Fitur Linux Container .....	27
Tabel 2.2.2 Daftar Perintah LXC .....	29
Tabel 2.6.1 Daftar Perintah Bridge-utils .....	32
Tabel 3.1.1 Penjelasan Diagram Kasus Penggunaan .....	33
Tabel 3.3.1 Daftar Rute pada <i>Backend</i> .....	37
Tabel 3.3.2 Daftar Rute pada <i>Frontend</i> .....	40
Tabel 3.3.3 Struktur Basis Data.....	43
Tabel 4.2.1 Implementasi Rute pada <i>Backend</i> .....	48
Tabel 4.3.1 Implementasi Peta Situs pada Antarmuka Praktikan .....	54
Tabel 4.4.1 Implementasi Peta Situs Antarmuka Admin .....	57
Tabel 4.5.1 Implementasi Basis Data .....	57
Tabel 5.2.1 Implementasi Uji Fungsionalitas .....	63
Tabel 5.3.1 Hasil Eksekusi Uji Fungsionalitas .....	69
Tabel 5.3.2 Waktu Respon Uji Fungsionalitas .....	71
Tabel 5.3.3 Hasil Uji Fungsionalitas dengan Skenario Modul Praktikum.....	73
Tabel 5.3.4 Hasil Uji Kapasitas dan Performa pada Komputer <i>Web Server</i> .....	76
Tabel 5.3.5 Hasil Uji Kapasitas dan Performa pada Komputer Pemandang .....	77

## DAFTAR TABEL

Tabel 2.2.1 Daftar Fitur Linux Container .....	27
Tabel 2.2.2 Daftar Perintah LXC .....	29
Tabel 2.6.1 Daftar Perintah Bridge-utils .....	32
Tabel 3.1.1 Penjelasan Diagram Kasus Penggunaan .....	33
Tabel 3.3.1 Daftar Rute pada <i>Backend</i> .....	37
Tabel 3.3.2 Daftar Rute pada <i>Frontend</i> .....	40
Tabel 3.3.3 Struktur Basis Data .....	43
Tabel 4.2.1 Implementasi Rute pada <i>Backend</i> .....	48
Tabel 4.3.1 Implementasi Peta Situs pada Antarmuka Praktikan .....	54
Tabel 4.4.1 Implementasi Peta Situs Antarmuka Admin .....	57
Tabel 4.5.1 Implementasi Basis Data .....	57
Tabel 5.2.1 Implementasi Uji Fungsionalitas .....	63
Tabel 5.3.1 Hasil Eksekusi Uji Fungsionalitas .....	69
Tabel 5.3.2 Waktu Respon Uji Fungsionalitas .....	71
Tabel 5.3.3 Hasil Uji Fungsionalitas dengan Skenario Modul Praktikum .....	73
Tabel 5.3.4 Hasil Uji Kapasitas dan Performa pada Komputer <i>Web Server</i> .....	76
Tabel 5.3.5 Hasil Uji Kapasitas dan Performa pada Komputer Pemandang .....	77

## DAFTAR KODE SUMBER

Kode Sumber 2.4.1 Contoh Implementasi Scapy dengan Python.....	30
Kode Sumber 2.4.2 Contoh Kode Program Kerangka Kerja Flask.....	30
Kode Sumber 2.5.1 Contoh Kode Program GoJS.....	31
Kode Sumber 3.3.1 Desain Paket Protokol ICMP tanpa Spoofing.....	44
Kode Sumber 3.3.2 Desain Paket Protokol ICMP dengan Spoofing.....	44
Kode Sumber 3.3.3 Desain Paket Protokol ICMP tanpa Flag	44
Kode Sumber 3.3.4 Desain Paket Protokol ICMP dengan Flag .....	45

# **BAB I**

## **PENDAHULUAN**

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

### **1.1 Latar Belakang**

Praktikum jaringan komputer merupakan kegiatan untuk menunjang mata kuliah jaringan komputer. Dalam praktikum jaringan komputer, praktikan diberi panduan untuk membuat simulasi dari jaringan komputer yang sebenarnya. Simulasi jaringan dalam praktikum jaringan komputer merupakan metode bagi praktikan untuk membuat topologi jaringan komputer tanpa harus memakai sumber daya yang sesungguhnya. Praktikum Jaringan Komputer yang sudah berlangsung di Jurusan Teknik Informatika dalam menyimulasikan topologi jaringan menggunakan User Mode Linux (UML). User Mode Linux merupakan aplikasi yang digunakan untuk menyimulasikan *router*, *host* dan *switch* dalam lingkungan virtual.

User Mode Linux mempunyai beberapa kekurangan, yaitu ketika menyimulasikan topologi dalam jumlah banyak yang menggunakan *host* dan *router*, *server* mengalami penurunan kinerja. Selain itu sumber daya penyimpanan di *server* juga berkurang karena besarnya ukuran *image* yang dibuat oleh UML untuk masing- masing *host* dan *router*. Hal ini dapat menyebabkan beberapa masalah seperti waktu *startup* yang lama, *latency* yang tinggi, dan kecenderungan *error*. Oleh karena itu dalam Tugas Akhir ini diusulkan lingkungan virtualisasi yaitu Linux Container dan simulasi pengiriman paket menggunakan Scapy.



Linux Container merupakan lingkungan virtual berbasis *container* yang menyediakan ukuran *image* yang lebih kecil dari User Mode Linux [1]. Proses *startup* pada Linux Container lebih cepat dibandingkan User Mode Linux meskipun banyak aplikasi *server* yang sudah dipasang. Dengan kelebihan Linux Container diatas, kecenderungan *error* yang diakibatkan oleh User Mode Linux dapat dikurangi.

Scapy [2] merupakan program untuk memanipulasi paket. Scapy dapat menata ulang sebuah paket dari beberapa protokol, mengirimkannya melalui jaringan, menganalisa paket tersebut, mencocokkan antara *request* dan *reply* . Scapy juga dapat digunakan untuk *scanning*, *tracerouting*, *probing*, *unit tests*, penyerangan atau *network discovery*. Sehingga dalam praktikum jaringan komputer, Scapy dapat dimanfaatkan untuk memperdalam praktikan dalam menganalisa paket secara detail apa saja yang dikirim dan respon yang diberikan oleh paket tersebut.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membangun aplikasi simulasi jaringan menggunakan Linux Container untuk praktikum jaringan komputer?
2. Bagaimana membuat antarmuka praktikum jaringan komputer menggunakan Scapy?
3. Bagaimana mendapat respon dari praktikum dalam mengerjakan tugas sebagai indikator evaluasi online?

## 1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Aplikasi simulasi ini terbatas hanya memanipulasi paket TCP/IP.
2. Sistem operasi yang dipakai sebagai image untuk LXC yaitu ubuntu.
3. Modul praktikum jaringan komputer yang akan diimplementasikan adalah modul *routing*, modul *web* dan *proxy server* , dan modul DNS dan DHCP *server* .

#### **1.4 Tujuan**

Tujuan dari dibuatnya Tugas Akhir adalah menghasilkan aplikasi simulasi jaringan dan simulasi manipulasi paket yang dapat digunakan dalam mata kuliah jaringan komputer yang memanfaatkan virtualisasi berbasis Linux Container dan manipulasi paket dengan Scapy.

#### **1.5 Manfaat**

Dengan dibuatnya Tugas Akhir ini akan memberikan konsep baru pada praktikum dalam membuat simulasi jaringan serta dapat membuat simulasi pengiriman paket TCP/IP yang sudah dimodifikasi sebelumnya.

#### **1.6 Metodologi**

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.  
Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan yang sama dengan Tugas Akhir ini, yaitu membuat aplikasi simulasi jaringan untuk praktikum jaringan komputer memanfaatkan Scapy dalam lingkungan lxc.
2. Studi literatur  
Pada tahap ini dilakukan pemahaman informasi dan literatur yang diperlukan untuk pembuatan implementasi.

Informasi dan literatur didapatkan dari literatur buku dan sumber-sumber informasi lain yang berhubungan.

3. Analisis dan desain perangkat lunak  
Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.
4. Implementasi perangkat lunak  
Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.
5. Pengujian dan evaluasi  
Pada tahapan ini dilakukan uji coba pada data yang telah dikumpulkan. Tahapan ini dimaksudkan untuk mengevaluasi kesesuaian data dan program serta mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.
6. Penyusunan buku Tugas Akhir.  
Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

### **1.7 Sistematika Penulisan Laporan Tugas Akhir**

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir secara keseluruhan. Selain itu,

diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

**Bab I      Pendahuluan**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

**Bab II     Tinjauan Pustaka**

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

**Bab III    Desain dan Perancangan**

Bab ini berisi tentang desain sistem yang disajikan dalam penjabaran diagram kasus penggunaan, diagram aktivitas, dan desain algoritma yang akan diimplementasikan.

**Bab IV    Implementasi**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode program yang digunakan untuk proses implementasi.

**Bab V     Uji Coba Dan Evaluasi**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

**Bab VI Kesimpulan Dan Saran**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

## BAB II

### TINJAUAN PUSTAKA

Pada bab ini berisi penjelasan teori-teori yang berkaitan dengan algoritma yang diajukan pada pengimplementasian program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### 2.1 Simulasi Jaringan

Simulasi jaringan [3] merupakan virtual reality simulation yang digunakan untuk menirukan tabiat dari proses dan sistem jaringan sesuai dengan dunia nyata. Berbagai percobaan dapat dilakukan dengan mengubah model pada simulasi. Penggunaan simulasi dapat membantu untuk menguji hal yang terlalu beresiko jika dilakukan secara nyata.

#### 2.2 Linux Container (LXC)

Linux Container (LXC) [4] adalah sebuah lingkungan virtualisasi *dilevel* sistem operasi yang dapat menjalankan sistem Linux standar yang terisolasi dalam sebuah *host* tanpa terpisah oleh *kernel host* tersebut. Tabel 2.2.1 merupakan daftar fitur yang diakomodasi oleh Linux Container.

**Tabel 2.2.1 Daftar Fitur Linux Container**

No	Nama Fitur	Keterangan
1	<i>Kernel namespaces(inter process communication, mount, pid, network, dan user)</i>	Virtualisasi yang dibuat oleh Linux Container memiliki lingkungan isolasi sendiri berbeda dengan komputer <i>host</i> , sehingga memiliki proses, jaringan, <i>user</i> ,

No	Nama Fitur	Keterangan
		dan <i>inter process communication</i> yang berbeda dengan komputer <i>host</i> .
2	Apparmor dan SELinux <i>profiles</i>	Fitur keamanan yang disediakan Linux Container.
3	Seccomp <i>policies</i>	Merupakan fitur dari <i>kernel</i> yang dapat menyaring <i>system call</i> yang biasa digunakan oleh proses dalam sistem operasi Linux beserta anak dari proses tersebut. Seccomp <i>policies</i> dapat ditambahkan pada virtualisasi oleh Linux Container.
4	CGroups	Pengguna dapat membatasi dan mengatur prioritas penggunaan sumber daya seperti <i>cpu</i> , <i>memory</i> , <i>block I/O</i> , dan <i>network</i> .

Linux Container terdiri dari beberapa komponen pengembangnya, antara lain *liblxc* sebagai pustaka khusus untuk *Linux Container* dan bahasa pemrograman yang digunakan untuk API antara lain python3, lua, go, dan ruby. Selain itu Linux Container juga menyediakan standar dalam mengontrol *container*-nya dan *template image* untuk *container*.

Tabel 2.2.2 merupakan daftar perintah dari *Linux Container* [5] yang digunakan dalam tugas akhir ini.

Tabel 2.2.2 Daftar Perintah LXC

No	Perintah	Contoh penggunaan	Keterangan
1	lxc-create	<i>lxc-create -n [nama kontainer] -t [nama template sistem operasi]</i>	Membuat kontainer baru.
2	lxc-clone	<i>lxc-clone -n [nama kontainer baru] -o [nama kontainer yang ingin disalin]</i>	Menyalin kontainer yang sudah tersedia.
3	lxc-start	<i>lxc-start -n [nama kontainer]</i>	Menghidupkan kontainer.
4	lxc-stop	<i>lxc-stop -n [nama kontainer]</i>	Mematikan kontainer.
5	lxc-destroy	<i>lxc-destroy -n [nama kontainer]</i>	Menghapus kontainer.
6	lxc-ls	<i>lxc-ls --fancy</i>	Menampilkan semua daftar kontainer yang sudah dibuat dengan statusnya.
7	lxc-info	<i>lxc-info -n [nama kontainer]</i>	Menampilkan informasi dari satu kontainer.

### 2.3 Scapy

Scapy [2] adalah program untuk memanipulasi sebuah paket. Scapy dapat menata ulang sebuah paket dari beberapa protokol, mengirimkannya melalui jaringan, menganalisa paket tersebut, dan mencocokkan antara *request* dan *reply*. Scapy dapat digunakan untuk *scanning*, *tracerouting*, *probing*, *unit tests*, penyerangan atau *network discovery*.

Scapy dibangun dengan bahasa pemrograman python. Kode Sumber 2.4.1 merupakan contoh implementasi Scapy menggunakan bahasa pemrograman python.



## 2.4 Flask

Flask [6] merupakan kerangka kerja mikro dari bahasa pemrograman python yang didukung oleh *werkzeug* dan *jinja2*. Aplikasi *web* yang dibuat dengan Flask disimpan dalam satu berkas *.py*.

Flask ingin menjadi *web* framework yang sederhana namun dapat diperluas dengan beragam pustaka tambahan yang sesuai dengan kebutuhan penggunaannya. Kode Sumber 2.4.2 merupakan contoh kode program kerangka kerja flask.

### Kode Sumber 2.4.1 Contoh Implementasi Scapy dengan Python

```
#!/usr/bin/env python
import sys
from scapy.all import sr1,IP,ICMP
p=sr1(IP(dst=sys.argv[1])/ICMP())
if p:
    p.show()
```

### Kode Sumber 2.4.2 Contoh Kode Program Kerangka Kerja Flask

```
from flask import Flask
app = Flask(__name__)
@app.route("/")
def hello():
    return "Hello World!"
if __name__ == "__main__":
    app.run()
```

## 2.5 GoJS

GoJS [7] merupakan pustaka dari *javascript* untuk mengimplementasikan diagram interaktif. Diagram yang mempunyai konstruksi *node* dan penghubung antar *node* yang sulit sekalipun dapat dibuat menggunakan GoJS dengan mudah karena mempunyai *template* dan *layout* yang beragam.

Selain itu, fitur yang didukung oleh GoJS antara lain *drag-and-drop*, *copy-and-paste*, *data binding and models*, *transactional state and undo management*, *palettes*, *overviews*, dan *event handlers*.

Dalam tugas akhir ini, GoJS digunakan sebagai tampilan untuk simulasi. Fitur yang dimanfaatkan dalam tugas akhir ini antara lain *template*, *data binding and models*, *drag and drop*, *palettes* dan *event handlers*. Kode Sumber 2.5.1 merupakan contoh kode program GoJS.

#### Kode Sumber 2.5.1 Contoh Kode Program GoJS

```
var $ = go.GraphObject.make;
var myDiagram =
  $(go.Diagram, "myDiagramDiv",
    {
      initialContentAlignment:
        go.Spot.Center, "undoManager.isEnabled": true });
var myModel = $(go.Model);
myModel.nodeDataArray = [
  { key: "Alpha" },
  { key: "Beta" },
  { key: "Gamma" }
];
myDiagram.model = myModel;
```

## 2.6 Bridge-utils

Bridge-utils [8] merupakan paket aplikasi yang digunakan untuk mengkonfigurasi *bridge* untuk *ethernet* di sistem operasi Linux. Bridge-utils dapat digunakan untuk menghubungkan perangkat *ethernet* sekaligus secara bersama-sama baik dari *host* yang mempunyai fisik nyata maupun berupa virtualisasi.

Dalam tugas akhir ini, Bridge-utils digunakan untuk mensimulasikan modul *switch* sesuai dengan keadaan di dunia nyata. Tabel 2.6.1 merupakan daftar perintah Bridge-utils.

**Tabel 2.6.1 Daftar Perintah Bridge-utils**

<b>No</b>	<b>Perintah</b>	<b>Keterangan</b>
1	<i>brctl addbr [nama bridge]</i>	Menambahkan <i>bridge</i> baru.
2	<i>brctl addif [nama bridge] [interface]</i>	Menambahkan <i>interface ethernet</i> ke dalam <i>bridge</i> yang sudah dibuat sebelumnya.
3	<i>brctl show</i>	Menampilkan daftar semua <i>bridge</i> yang sudah dibuat beserta <i>interface</i> yang terdapat dalam <i>bridge</i> tersebut.
4	<i>ifconfig [nama bridge] up</i>	Menghidupkan <i>bridge</i> yang sudah dibuat
5	<i>ifconfig [nama bridge] down</i>	Mematikan <i>bridge</i> yang sudah dibuat
6	<i>ifconfig [nama bridge] [alamat ip] netmask [netmask] up</i>	Menghidupkan <i>bridge</i> dengan mengatur alamat <i>ip</i> dan <i>netmask</i> .

## **BAB III**

### **DESAIN DAN PERANCANGAN**

Simulasi jaringan pada umumnya merupakan *virtual reality simulation* yang digunakan untuk menirukan tabiat dari proses dan sistem jaringan sesuai dengan dunia nyata. Fitur – fitur yang harus ada dalam simulasi jaringan adalah:

- Sistem dapat membuat pengguna baru dan memanipulasinya
- Sistem dapat membuat simulasi jaringan
- Sistem dapat membuat simulasi pengiriman paket
- Sistem dapat menampilkan hasil simulasi

Dari fitur umum tersebut maka dibuatlah analisis, perancangan dan implementasi yang dibahas pada bab ini.

#### **3.1 Kasus Penggunaan**

Gambar 3.2.1 menampilkan diagram kasus penggunaan sistem secara umum dengan penjelasan pada Tabel 3.1.1

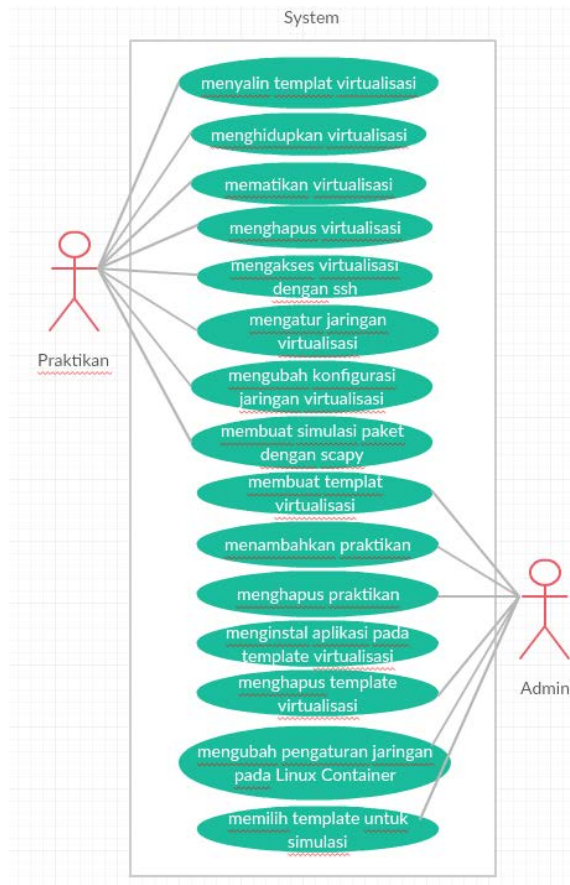
**Tabel 3.1.1 Penjelasan Diagram Kasus Penggunaan**

<b>No</b>	<b>Nama</b>	<b>Aktor</b>	<b>Deskripsi</b>
UC01	Menyalin <i>template</i> virtualisasi	Praktikan	Praktikan dapat menyalin <i>template</i> virtualisasi yang dibuat oleh admin.
UC02	Menghidupkan virtualisasi	Praktikan	Praktikan dapat menghidupkan virtualisasi yang telah dibuat.
UC03	Mematikan virtualisasi	Praktikan	Praktikan dapat mematikan virtualisasi yang telah dibuat.
UC04	Menghapus virtualisasi	Praktikan	Praktikan dapat menghapus virtualisasi yang telah dibuat.
UC05	Mengakses	Praktikan	Praktikan dapat

No	Nama	Aktor	Deskripsi
	virtualisasi dengan SSH		mengakses virtualisasi yang telah dibuat menggunakan SSH.
UC06	Mengatur jaringan virtualisasi	Praktikan	Praktikan dapat mengatur jaringan virtualisasi.
UC07	Mengubah konfigurasi jaringan virtualisasi	Praktikan	Praktikan dapat mengubah konfigurasi jaringan virtualisasi yang sudah diatur sebelumnya.
UC08	Membuat simulasi paket dengan Scapy	Praktikan	Praktikan dapat membuat simulasi paket dengan Scapy.
UC09	Membuat <i>template</i> virtualisasi	Admin	Admin dapat membuat <i>template</i> virtualisasi yang nantinya akan digunakan praktikan.
UC10	Menambahkan praktikan	Admin	Admin dapat menambahkan praktikan baru.
UC11	Menghapus praktikan	Admin	Admin dapat menghapus praktikan.
UC12	Menginstal aplikasi pada <i>template</i> virtualisasi	Admin	Admin dapat menginstal aplikasi pada <i>template</i> sesuai komponen.
UC13	Menghapus <i>template</i> virtualisasi	Admin	Admin dapat menghapus <i>template</i> virtualisasi yang telah dibuat.
UC14	Mengubah pengaturan jaringan Linux Container	Admin	Admin dapat merubah pengaturan jaringan pada Linux Container.
UC15	Memilih <i>template</i> untuk simulasi	Admin	Admin dapat memilih <i>template</i> yang digunakan dalam simulasi.

### 3.2 Deskripsi Fitur

Secara umum, fitur sistem dibagi menjadi dua: fitur untuk praktikan dan fitur untuk admin.



Gambar 3.2.1 Diagram Kasus Penggunaan

#### 3.2.1 Fitur untuk Praktikan

Fitur yang tersedia untuk praktikan :

- Praktikan dapat menyalin *template* virtualisasi yang dibuat oleh admin.

- Praktikan dapat menghidupkan virtualisasi yang telah dibuat.
- Praktikan dapat mematikan virtualisasi yang telah dibuat.
- Praktikan dapat menghapus virtualisasi yang telah dibuat.
- Praktikan dapat mengakses virtualisasi yang telah dibuat menggunakan SSH.
- Praktikan dapat mengatur jaringan virtualisasi.
- Praktikan dapat mengubah konfigurasi jaringan virtualisasi yang sudah diatur sebelumnya.
- Praktikan dapat membuat simulasi paket dengan Scapy.

### 3.2.2 Fitur untuk Admin

Fitur yang tersedia untuk admin:

- Admin dapat membuat *template* virtualisasi yang nantinya akan digunakan praktikan.
- Admin dapat menambahkan praktikan baru.
- Admin dapat menghapus praktikan.
- Admin dapat menginstal aplikasi pada *template* sesuai komponen.
- Admin dapat menghapus *template* virtualisasi yang telah dibuat.
- Admin dapat merubah pengaturan jaringan pada Linux Container.
- Admin dapat memilih template yang digunakan dalam simulasi.

## 3.3 Arsitektur Sistem

Gambar 3.3.1 merupakan desain arsitektur dari sistem yang akan dibuat. Arsitektur sistem meliputi *backend*, *frontend*, Linux Container dan Bridge-utils, basis data, dan manipulasi paket dengan Scapy.

### 3.3.1 Desain Umum Sistem

Sistem dibangun dealam beberapa komponen umum yang terkait satu sama lain, yaitu:

- **Backend** sebagai pengendali yang menghubungkan basis data dengan *frontend*.
- **Frontend** adalah antarmuka dari sistem yang berbasis *web*.
- **Linux Container dan Bridge-utils** sebagai lingkungan virtual yang dipanggil dengan bantuan *backend*. Digunakan sebagai lingkungan untuk menyimulasikan perangkat jaringan seperti *router*, *switch*, komputer dan *cloud*.
- **Basis data** sebagai tempat menyimpan segala macam data transaksi maupun data utama sistem.
- **Manipulasi paket dengan Scapy** sebagai aplikasi yang dapat memodifikasi paket untuk simulasi pengiriman paket oleh praktikan.

### 3.3.2 Desain Backend

*Backend* dibuat dengan kerangka kerja dari bahasa python yaitu Flask. *Backend* akan menangani segala bentuk operasi basis data yang dikirimkan dari *frontend*, memberikan data yang dibutuhkan *frontend*, dan mengirim perintah ke Linux Container sesuai dengan data dari *frontend*. Terdapat tiga jenis rute yang ditangani oleh *backend*, yaitu rute yang diakses tanpa autentikasi, rute yang hanya bisa diakses admin, dan rute yang hanya bisa diakses praktikan. Tabel 3.3.1 merupakan daftar rute pada *backend*. Rute yang dibuat dalam *backend*

**Tabel 3.3.1 Daftar Rute pada Backend**

No	Rute	Me- tode	Hak akses	Aksi
1	<i>/login</i>	POST	Tidak ada	Melakukan autentikasi <i>username</i> dan <i>password</i> .



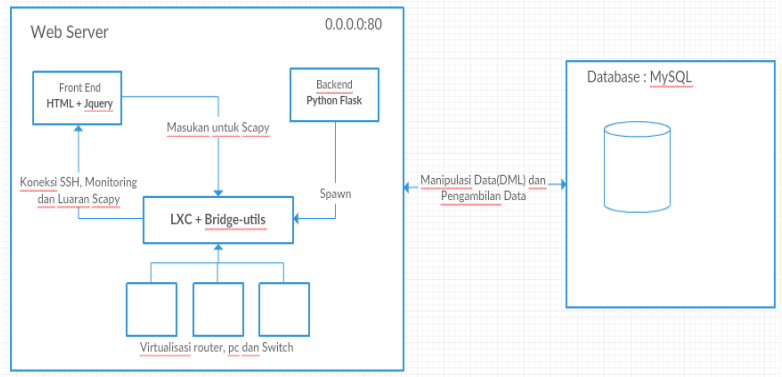
No	Rute	Me- tode	Hak akses	Aksi
2	<i>/lxc/create</i>	POST	Praktikan	Membuat virtualisasi baru dengan Linux Container untuk komponen <i>router</i> , komputer, <i>switch</i> dan <i>cloud</i> .
3	<i>/lxc/start</i>	POST	Praktikan	Menghidupkan virtualisasi Linux Container yang sudah dibuat.
4	<i>/lxc/stop</i>	POST	Praktikan	Mematikan virtualisasi Linux Container yang sudah dibuat.
5	<i>/lxc/setting</i>	POST	Praktikan	Melakukan pengaturan terhadap jaringan untuk virtual <i>cloud</i> , <i>router</i> , dan komputer.
6	<i>/lxc/delete</i>	POST	Praktikan	Menghapus komponen virtualisasi yang sudah dibuat
7	<i>/user/scapy</i>	POST	Praktikan	Membuat file berekstensi <i>.py</i> untuk simulasi paket Scapy menggunakan bahasa pemrograman python.
8	<i>/admin/create-template</i>	POST	Admin	Membuat <i>template</i> virtual <i>cloud</i> , <i>router</i> , dan pc yang nantinya akan disalin oleh praktikan.

No	Rute	Me- tode	Hak akses	Aksi
9	<i>/admin/adduser</i>	POST	Admin	Admin dapat menambahkan praktikan dalam sistem.
10	<i>/admin/deluser</i>	POST	Admin	Admin dapat menghapus praktikan dan admin ke dalam sistem.
11	<i>/admin/lxcnet</i>	POST	Admin	Admin dapat mengatur konfigurasi ulang pada jaringan Linux Container.
12	<i>/lxc/delete/&lt;template&gt;</i>	POST	Admin	Admin dapat menghapus <i>template</i> virtualisasi yang sudah dibuat.
13	<i>/lxc/instalapl</i>	POST	Admin	Admin dapat menginstal aplikasi untuk komponen <i>template</i> sesuai dengan aturan, yaitu <i>cloud</i> , <i>router</i> , dan komputer.
14	<i>/admin/settemplate</i>	POST	Admin	Admin dapat memilih <i>template</i> yang digunakan dalam simulasi.

### 3.3.3 Desain Frontend

Bagian *frontend* menggunakan HTML dan JQuery sedang untuk tampilan simulasi menggunakan pustaka *javascript* GoJS. Tabel 3.3.2 merupakan daftar rute yang ditangani oleh *frontend*. Gambar 3.3.2 menampilkan peta situs untuk halaman

praktikan pada dan peta situs untuk halaman admin pada Gambar 3.3.3.



**Gambar 3.3.1 Desain Arsitektur Sistem**

**Tabel 3.3.2 Daftar Rute pada *Frontend***

No	Rute	Me-tode	Hak Akses	Aksi
1	/	GET	Admin dan praktikan	Menampilkan halaman login pengguna.
2	/user/ dashboard	GET	Praktikan	Menampilkan halaman simulasi jaringan untuk praktikan.
3	/admin/ dashboard	GET	Admin	Menampilkan halaman kondisi komputer <i>host</i> yang dipakai untuk simulasi menggunakan Linux Container.
4	/admin/ template	GET	Admin	Menampilkan halaman pembuatan <i>template</i> Linux Container untuk komponen <i>router</i> ,

No	Rute	Me- tode	Hak Akses	Aksi
				pc, dan <i>cloud</i> .
5	<i>/admin/ usermanage</i>	GET	Admin	Menampilkan halaman untuk manajemen praktikan.
6	<i>/admin/ lxcnet</i>	GET	Admin	Menampilkan <i>form</i> konfigurasi Jaringan Linux Container

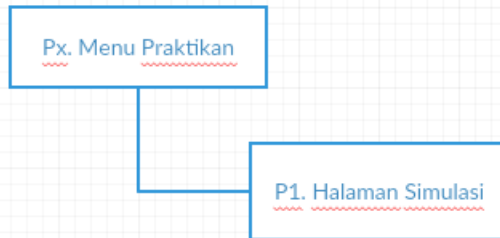
### 3.3.4 Desain Linux Container dan Bridge-utils

Linux Container digunakan sebagai lingkungan virtualisasi. Komponen yang divirtualisasikan adalah *router*, komputer(PC), dan *cloud*. Admin akan membuat *template* untuk masing-masing komponen. Sedangkan ketika melakukan simulasi, praktikan akan melakukan penyalinan dari *template* komponen yang sudah disediakan. Sistem operasi untuk komponen – komponen tersebut adalah Ubuntu.

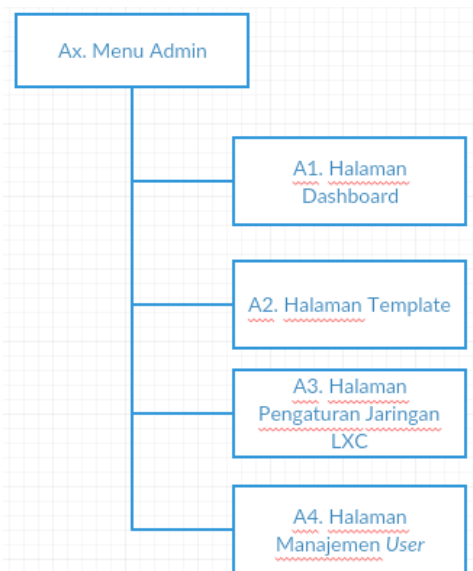
Komponen *cloud* digunakan untuk menghubungkan jaringan dalam lingkungan Linux Container dengan jaringan luar menggunakan virtual *switch*. Bridge-utils digunakan untuk virtual *switch* tersebut. Selain itu, Bridge-utils juga digunakan untuk menghubungkan komponen-komponen lain yaitu komputer dan *router*. Dalam komponen *cloud* diinstal aplikasi shellinabox, aplikasi yang dapat menampilkan terminal melalui peramban, untuk mengakses virtualisasi lain menggunakan koneksi SSH. Untuk melakukan simulasi Scapy, diinstal aplikasi Scapy di dalam komponen komputer.

Komponen komputer dan *router* dalam lingkungan Linux Container menggunakan alamat IP *private* sedangkan komponen *cloud* menggunakan IP yang terhubung dari jaringan luar dan IP *private* yang terhubung dengan jaringan di

lingkungan Linux Container. Gambar 3.3.4 menampilkan desain komponen virtualisasi.



**Gambar 3.3.2 Peta Situs Halaman Praktikan**



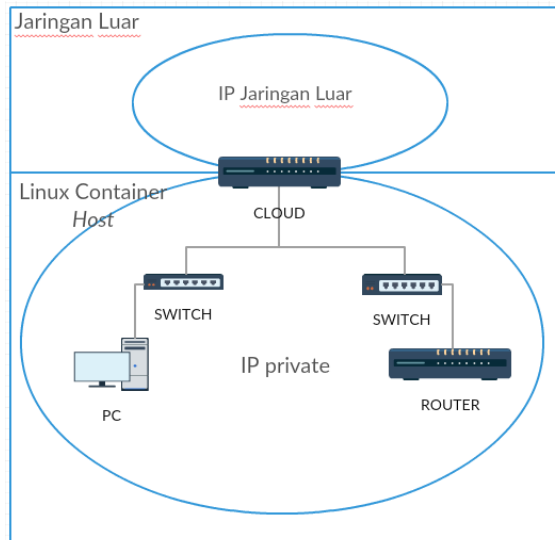
**Gambar 3.3.3 Peta Situs Halaman Admin**

### 3.3.5 Desain Basis Data

Basis data dari sistem menggunakan MySQL. Tabel 3.3.3 merupakan struktur basis data dari sistem. Terdapat satu tabel *users* yang menyimpan data dari pengguna.

Tabel 3.3.3 Struktur Basis Data

No	Nama Tabel	Deskripsi
1	users	Menyimpan data pengguna dan json hasil simulasi
2	template	Menyimpan data <i>template</i> komponen virtualisasi



Gambar 3.3.4 Desain Komponen Virtualisasi

### 3.3.6 Desain Manipulasi Paket dengan Scapy

Scapy adalah program untuk memanipulasi sebuah paket. Praktikan akan mendefinisikan paket terlebih dahulu melalui antarmuka *web*, kemudian sistem akan membuat sebuah file berekstensi *.py* yang berisi paket tersebut di dalam komponen virtual komputer. Praktikan dapat mensimulasikan pengiriman paket dengan Scapy melalui komponen virtual computer. Kode Sumber 3.3.1 merupakan desain dari paket dengan protokol ICMP yang memuat alamat IP tujuan target dan *payload* yang akan dikirimkan oleh paket tersebut.

**Kode Sumber 3.3.1 Desain Paket Protokol ICMP tanpa Spoofing**

```
#!/usr/bin/env python
import sys
from scapy.all import sr1,IP,ICMP
paket=sr1(IP(dst=[a][1])/ICMP()/[b])
print paket
```

*Keterangan: a = alamat ip tujuan, b = payload*

Kode Sumber 3.3.2 merupakan desain paket yang sama dengan paket pada Kode Sumber 3.3.1, namun terdapat alamat IP sumber yang mana dapat diisi sesuai keadaan sebenarnya atau alamat IP palsu.

**Kode Sumber 3.3.2 Desain Paket Protokol ICMP dengan Spoofing**

```
#!/usr/bin/env python
import sys
from scapy.all import sr1,IP,ICMP
paket=sr1(IP(src=[a]dst=[b][1])/ICMP()/[c])
print paket
```

*Keterangan: a = alamat ip sumber atau yang lain,  
b = alamat ip tujuan, c payload*

Kode Sumber 3.3.3 merupakan desain dari paket dengan protokol TCP yang memuat alamat IP tujuan dan *port* tujuan.

**Kode Sumber 3.3.3 Desain Paket Protokol ICMP tanpa Flag**

```
#!/usr/bin/env python
import sys
from scapy.all import sr1,IP,TCP
paket=sr1(IP(dst=[a][1])/TCP(dport=[b]))
print paket
```

*Keterangan: a = alamat ip tujuan, b = port tujuan*

Kode Sumber 3.3.4 merupakan desain dari paket dengan protokol TCP yang memuat alamat IP tujuan dan port tujuan. Dalam paket ini ditambahkan *flag* berupa ACK atau SYN ACK.

**Kode Sumber 3.3.4 Desain Paket Protokol ICMP dengan Flag**

```
#!/usr/bin/env python
import sys
from scapy.all import sr1,IP,TCP
paket=sr1(IP(dst=[a][1])/TCP(dport=[b],flags=[c]))
print paket
```

*Keterangan: a = alamat ip tujuan, b = port tujuan, c = flag*



*[Halaman ini sengaja dikosongkan]*

## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

### **4.1 Lingkungan Implementasi**

Lingkungan implementasi dan pengembangan dilakukan menggunakan komputer Laboratorium Arsitektur dan Jaringan Komputer dengan spesifikasi Intel *Core* I3 2120 dan memori 8GB. Perangkat lunak yang digunakan dalam proses pengembangan antara lain:

- Sistem operasi Ubuntu Linux *Server* 14.04.3 LTS
- Editor teks Sublime Text 3
- Flask 0.10.1 untuk kerangka kerja pemrograman
- LXC 1.0.8 untuk lingkungan virtualisasi
- Scapy 2.2.0 untuk simulasi paket
- GoJS v1.5.10 untuk tampilan simulasi
- Samba Version 4.1.6-Ubuntu
- Peramban *web* Mozilla Firefox

### **4.2 Implementasi *Backend***

Implementasi *backend* dengan kerangka kerja Flask 0.10.1. Rute yang terdapat pada *backend* terdapat pada Tabel 3.3.1. Tabel 4.2.1 merupakan penjelasan implementasi rute pada *backend*. Rute *backend* diimplementasikan dalam sebuah file python. Kode sumber implementasi *backend* terdapat pada lampiran.

Tabel 4.2.1 Implementasi Rute pada *Backend*

No	Rute	Hak Akses	Masukan	Luaran	Langkah proses
1	POST <i>/login</i>	Praktikan, Admin	<i>Username, password</i>	Halaman dashboard untuk masing-masing hak akses	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi <i>username</i> dan <i>password</i>.</li> <li>2. Jika praktikan atau admin terdaftar <ul style="list-style-type: none"> <li>• Admin akan diarahkan ke halaman admin dengan rute <i>/admin/dashboard</i>.</li> <li>• Praktikan akan diarahkan ke halaman simulasi dengan rute <i>/user/dashboard</i>.</li> </ul> </li> </ol>
2	POST <i>/lxc/create</i>	Praktikan	Nama komponen virtualisasi	Notifikasi	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi nama komponen virtualisasi.</li> <li>2. Melakukan perintah <i>lxc-clone</i> pada <i>template</i> berdasarkan data nama komponen yang diberikan.</li> <li>3. Mengirim data berupa json ke <i>frontend</i> ketika proses penyalinan selesai dilakukan.</li> </ol>
3	POST <i>/lxc/start</i>	Praktikan	Nama komponen virtualisasi	Notifikasi	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi nama komponen virtualisasi.</li> <li>2. Melakukan perintah <i>lxc-start</i> untuk menghidupkan komponen virtualisasi.</li> </ol>

No	Rute	Hak Akses	Masukan	Luaran	Langkah proses
					3. Mengirim data berupa json ke <i>frontend</i> ketika proses menghidupkan komponen virtualisasi selesai dilakukan.
4	POST <i>/lxc/stop</i>	Praktikan	Nama komponen virtualisasi	Notifikasi	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi nama komponen virtualisasi.</li> <li>2. Melakukan perintah <i>lxc-stop</i> untuk mematikan komponen virtualisasi.</li> <li>3. Mengirim data berupa json ke <i>frontend</i> ketika proses mematikan komponen virtualisasi selesai dilakukan.</li> </ol>
5	POST <i>/lxc/seting</i>	Praktikan	<i>Interface</i> , alamat ip, netmask, alamat gateway, nama switch	Notifikasi	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi <i>interface</i>, alamat ip, netmask, alamat gateway, nama switch</li> <li>2. Melakukan perubahan pada file <i>/etc/network/interfaces</i> pada komponen virtualisasi dan file <i>/var/lib/lxc/&lt;username&gt;/&lt;namakomponen&gt;/config</i> pada <i>host</i> sesuai dengan data yang diterima.</li> <li>3. Mengirim data berupa json ke</li> </ol>

No	Rute	Hak Akses	Masukan	Luaran	Langkah proses
					<i>frontend</i> ketika proses perubahan pada file di komponen virtualisasi dan file konfigurasi pada <i>host</i> selesai dilakukan.
6	POST <i>/lxc/delete</i>	Praktikan	Nama <i>template</i>	Notifikasi	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi <i>interface</i>, alamat ip, netmask, alamat <i>gateway</i>, nama <i>switch</i>.</li> <li>2. Melakukan perintah <i>lxc-destroy</i> untuk membuat <i>template</i> baru dengan masukan nama <i>template</i>.</li> <li>3. Mengirim data berupa json ke <i>frontend</i> ketika proses penghapusan <i>template</i> komponen virtualisasi selesai dilakukan.</li> </ol>
7	POST <i>/user/Scapy</i>	Praktikan	Nama protokol, alamat tujuan, <i>port</i> tujuan, <i>payload</i> ,	Notifikasi berisi lokasi penyimpanan file Scapy.	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi nama protokol, alamat tujuan, <i>port</i> tujuan, <i>payload</i>, <i>flag</i>.</li> <li>2. Melakukan pembuatan file baru yang berisi kode program berbahasa python untuk menjalankan Scapy sesuai dengan format Scapy dan data yang</li> </ol>

No	Rute	Hak Akses	Masukan	Luaran	Langkah proses
			<i>flag</i>		diterima dari <i>frontend</i> pada lokasi <i>/home/ubuntu</i> pada komponen virtualisasi PC. 3. Mengirim data berupa json yang berisi nama file yang telah dibuat.
8	POST <i>/admin/create-template</i>	Admin	Nama <i>template</i> , nama sistem operasi <i>template</i>	Notifikasi	1. Menerima data dari frontend berisi nama <i>template</i> , nama sistem operasi <i>template</i> 2. Melakukan perintah <i>lxc-create</i> untuk membuat <i>template</i> baru dengan masukan Nama <i>template</i> , nama sistem operasi <i>template</i> . 3. Mengirim data berupa json ke <i>frontend</i> ketika proses pembuatan <i>template</i> komponen virtualisasi selesai dilakukan.
9	POST <i>/adduser</i>	Admin	<i>Username</i> , <i>password</i>	-	1. Menerima data dari frontend berisi <i>username</i> dan <i>password</i> . 2. Mengecek apakah sudah terdaftar atau belum 3. Apabila belum terdaftar, mengirim

No	Rute	Hak Akses	Masukan	Luaran	Langkah proses
					perintah ke database untuk menambahkan data ke dalam database.
10	POST <i>/admin/deluser</i>	Admin	<i>Username</i>	-	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi <i>username</i> yang ingin dihapus.</li> <li>2. Mengirim perintah ke database untuk menghapus data sesuai dengan <i>username</i>.</li> </ol>
11	POST <i>/admin/lxcnet</i>	Admin	Nama <i>switch</i> , alamat IP, <i>netmask</i> , <i>network</i>	Notifikasi	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi nama <i>switch</i>, alamat IP, <i>netmask</i>, <i>network</i>.</li> <li>2. Mengirim perintah ke sistem operasi untuk melakukan pengaturan ulang pada jaringan Linux Container.</li> </ol>
12	POST <i>/lxc/delete/</i> <i>&lt;template&gt;</i>	Admin	Nama <i>template</i>	Notifikasi	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi nama <i>template</i> yang ingin dihapus.</li> <li>2. Melakukan perintah <i>lxc-destroy</i> untuk membuat <i>template</i> baru dengan data masukan dari <i>frontend</i> berisi nama <i>template</i>.</li> <li>3. Mengirim data berupa json ke <i>frontend</i> ketika proses penghapusan <i>template</i></li> </ol>

No	Rute	Hak Akses	Masukan	Luaran	Langkah proses
					komponen virtualisasi selesai dilakukan.
13	POST <i>/lxc/instalapl</i>	Admin	Nama aplikasi	Notifikasi	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi nama aplikasi yang akan diinstal.</li> <li>2. Mengirim perintah ke sistem operasi untuk menginstal aplikasi pada <i>template</i> virtualisasi.</li> </ol>
14	POST <i>/admin/settemplate</i>	Admin	Nama komponen <i>template</i>	Notifikasi	<ol style="list-style-type: none"> <li>1. Menerima data dari frontend berisi Nama komponen <i>template</i> yang akan digunakan untuk simulasi</li> <li>2. Mengirim perintah ke database untuk merubah nama template sesuai dengan aturan komponen <i>template</i> virtualisasi.</li> </ol>

### 4.3 Implementasi *Frontend*

*Frontend* adalah antarmuka untuk pengguna sistem. *Frontend* dibangun menggunakan HTML dan JQuery sedang untuk tampilan simulasi menggunakan pustaka *javascript GoJS*.

#### 4.3.1 Antarmuka Praktikan

Terdapat delapan kasus penggunaan dalam antarmuka praktikan, yaitu :

- Praktikan dapat menyalin *template* virtualisasi yang dibuat oleh admin.



- Praktikan dapat menghidupkan virtualisasi yang telah dibuat.
- Praktikan dapat mematikan virtualisasi yang telah dibuat.
- Praktikan dapat menghapus virtualisasi yang telah dibuat.
- Praktikan dapat mengakses virtualisasi yang telah dibuat menggunakan SSH.
- Praktikan dapat mengatur jaringan virtualisasi.
- Praktikan dapat mengubah konfigurasi jaringan virtualisasi yang sudah diatur sebelumnya.
- Praktikan dapat membuat simulasi paket dengan Scapy.

Tabel 4.3.1 merupakan implementasi peta situs pada antarmuka praktikan.

**Tabel 4.3.1 Implementasi Peta Situs pada Antarmuka Praktikan**

No	Halaman	Rute	Use Case	Operasi
1	Px dan P1	/user/dashboard	UC01	1. Melakukan <i>drag</i> pada komponen <i>template</i> 2. Melakukan <i>drop</i> komponen yang sedang <i>didrag</i> pada kolom simulasi Menyalin komponen virtualisasi
			UC02	1. Klik kanan komponen pada kolom simulasi 2. Pilih menu <b>Hidupkan</b> Menghidupkan virtualisasi yang telah dibuat
			UC03	1. Klik kanan komponen pada kolom simulasi 2. Pilih menu <b>Matikan</b>

No	Halaman	Rute	Use Case	Operasi
				Mematikan virtualisasi yang telah dibuat
			UC04	1. Klik kanan komponen pada kolom simulasi 2. Pilih menu <b>Hapus</b> Menghapus virtualisasi yang telah dibuat
			UC05	1. Klik kanan komponen pada kolom simulasi 2. Pilih menu <b>SSH Terminal</b> Mengakses virtualisasi yang telah dibuat
			UC06	1. Klik kanan komponen pada kolom simulasi 2. Pilih menu <b>Atur Jaringan</b> Mengatur jaringan virtualisasi yang telah dibuat
			UC07	1. Pilih menu <b>Ubah</b> pada tabel dibawah kolom simulasi 2. Mengubah konfigurasi jaringan yang telah diatur sebelumnya
			UC08	1. Klik kanan komponen pada kolom simulasi 2. Pilih menu Manipulasi Paket 3. Membuat simulasi paket dengan Scapy

#### 4.3.2 Antarmuka Admin

Terdapat enam kasus penggunaan dalam antarmuka admin, yaitu:

- Admin dapat membuat *template* virtualisasi yang nantinya akan digunakan praktikan.
- Admin dapat memantau virtualisasi yang dibuat oleh praktikan.
- Admin dapat menambahkan praktikan baru.

- Admin dapat menghapus praktikan.
- Admin dapat menambahkan admin baru.
- Admin dapat menambahkan admin.
- Admin dapat memilih template yang digunakan dalam simulasi.

#### 4.4 Implementasi Linux Container dan Bridge-utils

Pada sub-bab ini akan dibahas mengenai implementasi Linux Container dan bridge-utils yang digunakan dalam sistem. Tabel 4.2.1 merupakan implementasi rute pada *backend* yang menangani hubungan antara *frontend* dengan Linux Container dan bridge-utils. Terdapat 3 komponen *template* virtual yaitu *router*, *cloud*, dan komputer untuk layanan simulasi. Komponen tersebut dibuat oleh admin, kemudian praktikan akan melakukan penyalinan pada *template* tersebut ketika melakukan simulasi.

Fitur Linux Container yang disediakan dalam simulasi adalah praktikan dapat menghidupkan komponen virtual, praktikan dapat mematikan komponen virtual, praktikan dapat mengonfigurasi jaringan pada komponen virtual, dan praktikan dapat menggunakan virtual melalui *ssh* dengan bantuan aplikasi shellinabox yang terinstal pada komponen *cloud*. Untuk dapat terhubung antar komponen, dipakailah bridge-utils sebagai virtual *switch*. Komponen akan tersambung dengan virtual *switch* apabila sudah melakukan konfigurasi jaringan.

Gambar 4.5.1 merupakan implementasi dari komponen virtualisasi di mana komponen *cloud* akan menghubungkan komponen yang berada dalam lingkungan Linux Container dengan jaringan di Laboratorium Arsitektur dan Jaringan Komputer.

**Tabel 4.4.1 Implementasi Peta Situs Antarmuka Admin**

No	Halaman	Rute	Use Case	Operasi
1	Ax dan A1	/admin/ dashboard	-	-
2	A2	/admin/ template	UC09, UC12, UC13, UC15	<p><b>Create template</b> : membuat <i>template</i> baru untuk komponen virtualisasi.</p> <p><b>Instal</b> : menginstal aplikasi pada komponen <i>template</i>.</p> <p><b>Hapus</b> : menghapus komponen <i>template</i>.</p> <p><b>Pilih Template</b> : memilih <i>template</i> untuk digunakan praktikan dalam simulasi.</p>
3	A4	/admin /usermanage	UC10, UC11,	<p><b>Tambah Praktikan</b> : menambahkan praktikan</p> <p><b>Hapus</b> : Menghapus praktikan.</p>
5	A3	/admin/ lxcnet	UC14	<b>Update</b> : merubah konfigurasi jaringan Linux Container

#### 4.5 Implementasi Basis Data

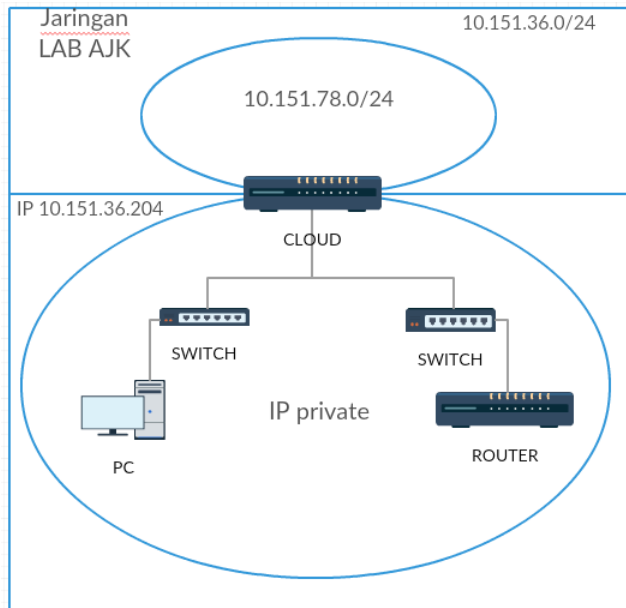
Pada sub-bab ini akan dibahas mengenai detail struktur basis data yang digunakan dalam sistem. Tabel 4.5.1 merupakan detail implementasi dari basis data pada sistem. Detail meliputi nama atribut, tipe data dan deskripsi masing-masing kolom.

**Tabel 4.5.1 Implementasi Basis Data**

No	Nama Tabel	Nama Kolom	Tipe Data	Deskripsi
1	users	userid	Integer	<i>Primary key</i>

No	Nama Tabel	Nama Kolom	Tipe Data	Deskripsi
				tabel
2	users	username	Varchar(255)	Nama pengguna
3	users	userpass	Varchar(255)	Password pengguna
4	users	savedsim	text	Json simulasi terakhir sebelum praktikan keluar dari sistem.
5	users	savedlink	text	Json hubungan antar komponen virtual terakhir sebelum praktikan keluar dari sistem.
6	users	su	Integer	Hak akses(0. Praktikan, 1. Admin)
7	users	login	Integer	<i>Flag</i> untuk mengecek apakah pengguna sudah login atau belum di browses yang berbeda (0. belum login 1. login)
8	template	templateid	Integer	<i>Primary key</i> tabel
9	template	templatrouter	Varchar(255)	Nama template untuk aturan komponen <i>cloud</i> .
10	template	templatecloud	Varchar(255)	Nama template untuk aturan komponen <i>router</i> .

No	Nama Tabel	Nama Kolom	Tipe Data	Deskripsi
11	template	templatepc	Varchar(255)	Nama template untuk aturan komponen komputer.



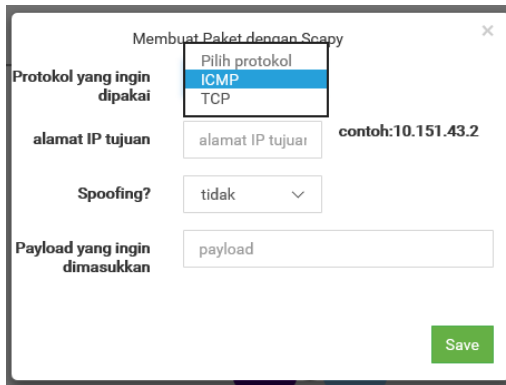
**Gambar 4.5.1 Implementasi Komponen Virtualisasi**

#### 4.6 Implementasi Manipulasi Paket dengan Scapy

Pada sub-bab ini akan dibahas mengenai implementasi Scapy yang digunakan dalam sistem. Paket yang akan dibuat dengan menggunakan Scapy adalah paket dengan protokol icmp dan tcp. Praktikan membuat paket dengan menekan klik kanan pada komponen komputer. Sistem membuat file berekstensi .py dilokasi /home/ubuntu pada komponen komputer ketika praktikan sudah selesai membuat paket dan sistem akan menampilkan notifikasi nama file yang sudah

dibuat. Gambar 4.6.1 merupakan tampilan *form* manipulasi paket protokol ICMP, sedangkan Gambar 4.6.2 merupakan tampilan *form* manipulasi paket protokol TCP.

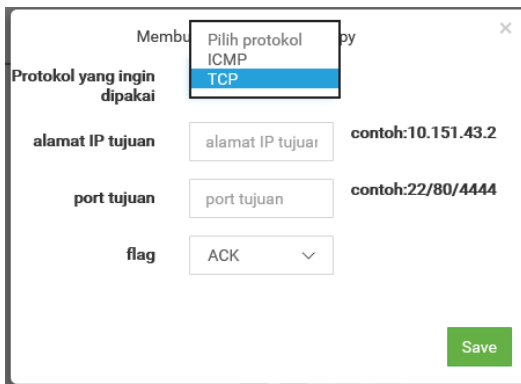
Untuk melakukan simulasi pengiriman paket, praktikan melakukan *ssh* kekomponen komputer melalui cloud, kemudian praktikan menjalankan file python yang sudah dibuat dengan kondisi sebagai *root*. Setelah dijalankan, file tersebut akan memberi nilai keluaran hasil dari pengiriman paket yang sudah dilakukan.



The screenshot shows a web form titled "Membuat Paket dengan Scapy". It contains the following fields and options:

- Protokol yang ingin dipakai:** A dropdown menu with "Pilih protokol" at the top, and "ICMP" and "TCP" as options. "ICMP" is currently selected.
- alamat IP tujuan:** A text input field with the placeholder "alamat IP tujuan" and an example value "contoh:10.151.43.2".
- Spoofing?:** A dropdown menu with "tidak" and a downward arrow.
- Payload yang ingin dimasukkan:** A text input field with the placeholder "payload".
- Save:** A green button at the bottom right.

**Gambar 4.6.1 Tampilan *Form* Paket Protokol ICMP**



The screenshot shows the same web form "Membuat Paket dengan Scapy" but configured for the TCP protocol. The fields and options are:

- Protokol yang ingin dipakai:** A dropdown menu with "Pilih protokol" at the top, and "ICMP" and "TCP" as options. "TCP" is currently selected.
- alamat IP tujuan:** A text input field with the placeholder "alamat IP tujuan" and an example value "contoh:10.151.43.2".
- port tujuan:** A text input field with the placeholder "port tujuan" and an example value "contoh:22/80/4444".
- flag:** A dropdown menu with "ACK" and a downward arrow.
- Save:** A green button at the bottom right.

**Gambar 4.6.2 Tampilan *Form* Paket Protokol TCP**

## **BAB V**

### **UJI COBA DAN EVALUASI**

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba yang meliputi uji kebenaran dan uji kinerja serta analisa setiap pengujian.

#### **5.1 Lingkungan Uji Coba**

Lingkungan untuk pengujian menggunakan tiga buah komputer yang terdiri dari : satu *web server* , satu komputer penguji, dan satu komputer untuk pembandingan uji kapasitas dan performa. *Web server* diletakkan pada komputer sama yang digunakan untuk implementasi. Sementara dua komputer lainnya di luar dari proses implementasi.

Proses pengujian dilakukan di laboratorium Arsitektur dan Jaringan Komputer gedung Teknik Informatika ITS. Spesifikasi perangkat lunak dan perangkat keras masing-masing komputer sebagai berikut:

- *Web Server* (IP 10.151.36.204)
  - Perangkat Keras
    - ✓ Komputer fisik
    - ✓ Prosesor Intel *Core* I3 2120
    - ✓ RAM 8GB
    - ✓ Hardisk 500 GB
  - Perangkat Lunak
    - ✓ *Ubuntu Server* Linux 14.02.03 LTS
    - ✓ Flask 0.10.1
    - ✓ LXC 1.0.8
    - ✓ Scapy 2.2.0
    - ✓ Python 2.7
- Penguji Fungsionalitas



- Perangkat Keras
  - ✓ Komputer fisik
  - ✓ Prosesor Intel *Core* I3 2120
  - ✓ RAM 8 GB
  - ✓ Hardisk 500 GB
- Perangkat Lunak
  - ✓ Windows 8
  - ✓ Mozilla Firefox
- Pembandingan
  - Perangkat Keras
    - ✓ Komputer fisik
    - ✓ Prosesor Intel(R) *Core*(TM)2 Duo CPU E7400 @ 2.80GHz
    - ✓ RAM 2 GB
    - ✓ Hardisk 320 GB
  - Perangkat Lunak
    - ✓ Ubuntu *Server* Linux 14.02.03 LTS
    - ✓ LXC 1.0.8
    - ✓ Python 2.7
- Penguji Kapasitas dan Performa
  - Perangkat Keras
    - ✓ Komputer fisik 10 buah
    - ✓ Prosesor Intel(R) *Core*(TM)2 Duo CPU E7400 @ 2.80GHz
    - ✓ RAM 2 GB
    - ✓ Hardisk 320 GB
  - Perangkat Lunak
    - ✓ Windows 7
    - ✓ Mozilla Firefox
    - ✓ Chrome

## 5.2 Skenario Uji Coba

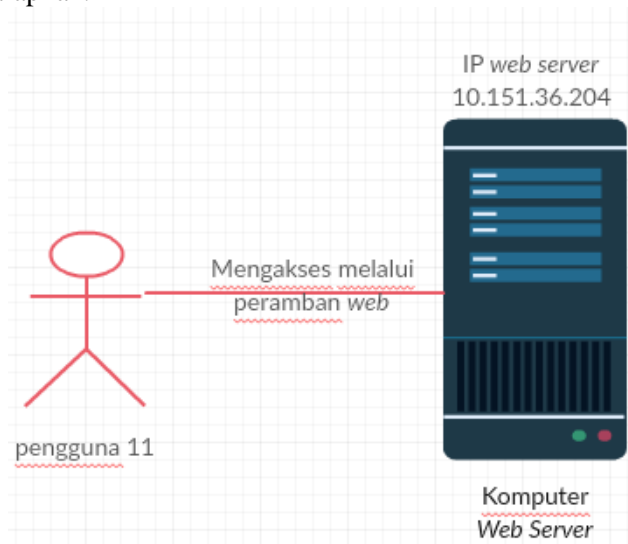
Skenario uji coba dilakukan dalam beberapa tahap uji coba:

- Uji Fungsionalitas

- Uji Fungsionalitas dengan Skenario Modul Praktikum
- Uji Kapasitas dan Performa

### 5.2.1 Uji Fungsionalitas

Uji dilakukan dengan melakukan uji coba membuka aplikasi melalui peramban *web*. Gambar 5.2.1 merupakan desain arsitektur uji fungsionalitas. Uji coba ini berguna untuk menguji apakah operasi-operasi dasar yang dilakukan dapat memberikan hasil yang diharapkan. Peramban *web* yang digunakan adalah Mozilla Firefox. Tabel 5.2.1 menunjukkan rancangan setiap aksi pengujian dan hasil atau nilai yang diharapkan.



**Gambar 5.2.1 Desain Arsitektur Uji Fungsionalitas**

**Tabel 5.2.1 Implementasi Uji Fungsionalitas**

No	Nama Kegiatan	Uji Coba	Hasil Harapan
1	<i>Login</i> sebagai	Dapat <i>login</i> ke	Pengguna dapat

No	Nama Kegiatan	Uji Coba	Hasil Harapan
	admin	dalam sistem dengan <i>username</i> dan <i>password</i> yang sudah terdaftar	memasuki halaman sesuai <i>role</i> .
2	<i>Login</i> sebagai praktikan	Dapat <i>login</i> ke dalam sistem dengan <i>username</i> dan <i>password</i> yang sudah terdaftar	Pengguna dapat memasuki halaman sesuai <i>role</i> .
3	Menyalin <i>template</i> virtualisasi	Dapat men- <i>drag template</i> dan men- <i>drop</i> pada kolom simulasi.	<i>Template</i> virtualisasi tersalin dan notifikasi muncul dihalaman simulasi.
4	Menghidupkan virtualisasi	Dapat menghidupkan virtualisasi	Komponen virtualisasi hidup dan notifikasi muncul dihalaman simulasi.
5	Mematikan virtualisasi	Dapat mematikan virtualisasi	Komponen virtualisasi mati dan notifikasi muncul dihalaman simulasi.
6	Menghapus virtualisasi	Dapat menghapus virtualisasi	Komponen virtualisasi terhapus dan notifikasi muncul dihalaman simulasi.
7	Mengakses virtualisasi dengan SSH	Dapat mengakses virtualisasi	Layar baru pada peramban muncul dengan memuat halaman dengan alamat berupa IP <i>cloud</i> dengan <i>port</i> 4200.
8	Mengatur	Dapat mengatur	Pengaturan jaringan

No	Nama Kegiatan	Uji Coba	Hasil Harapan
	jaringan virtualisasi	jaringan pada komponen virtualisasi	tersimpan
9	Mengubah konfigurasi jaringan virtualisasi	Dapat mengatur ulang jaringan pada komponen virtualisasi	Pengaturan ulang jaringan tersimpan
10	Membuat simulasi paket ICMP dengan Scapy	Dapat membuat berkas berekstensi .py yang berisi kode program Scapy	Berkas berada di dalam <i>folder /home/ubuntu</i> pada komponen komputer dan muncul notifikasi berupa nama berkas yang telah dibuat.
11	Membuat simulasi paket TCP dengan Scapy	Dapat membuat berkas berekstensi .py yang berisi kode program Scapy	Berkas berada di dalam <i>folder /home/ubuntu</i> pada komponen komputer dan muncul notifikasi berupa nama berkas yang telah dibuat.
12	Membuat <i>template</i> virtualisasi	Dapat membuat <i>template</i> virtualisasi	<i>Template</i> ditambahkan
13	Menambahkan praktikan	Dapat menambahkan praktikan ke dalam sistem.	Praktikan ditambahkan.
14	Menghapus praktikan	Dapat menghapus praktikan dari sistem.	Praktikan terhapus.
15	Menginstal aplikasi pada <i>template</i> virtualisasi	Dapat menginstal aplikasi pada <i>template</i>	Aplikasi terinstal dan notifikasi muncul dihalaman simulasi.

No	Nama Kegiatan	Uji Coba	Hasil Harapan
		virtualisasi	
16	Menghapus <i>template</i> virtualisasi	Dapat menghapus <i>template</i> virtualisasi	<i>Template</i> terhapus dan notifikasi muncul dihalaman simulasi.
17	Mengubah pengaturan jaringan Linux Container	Dapat mengubah pengaturan jaringan Linux Container	Pengaturan berubah dan notifikasi muncul dihalaman simulasi.
18	Memilih <i>template</i> untuk simulasi	Dapat memilih <i>template</i> untuk simulasi	<i>Template</i> berubah dan notifikasi muncul dihalaman simulasi.

### 5.2.2 Uji Fungsionalitas dengan Skenario Modul Praktikum

Uji dilakukan oleh pengguna dari mahasiswa Teknik Informatika ITS dengan melakukan uji coba sesuai dengan modul praktikum jaringan komputer. Modul yang akan digunakan sebagai uji coba adalah modul *web server*. Modul tersebut membutuhkan sebuah komponen *cloud*, sebuah komponen komputer dan sebuah komponen *switch*. Gambar 5.2.2 merupakan skenario pengujian menggunakan modul *web server*.

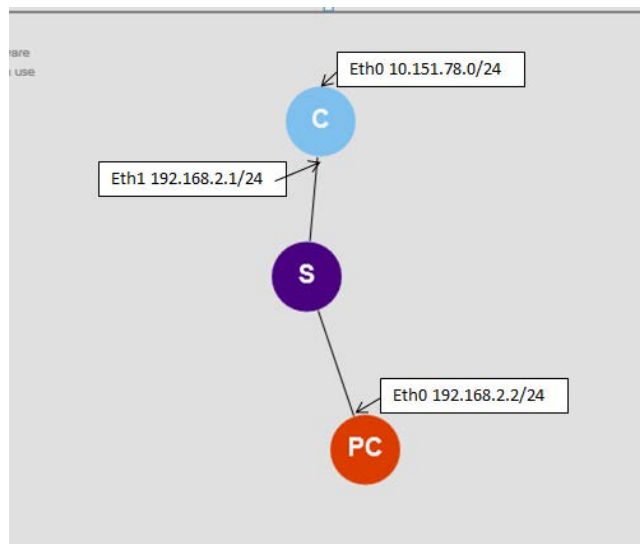
*Web server* akan diinstal pada komponen *cloud* dan aplikasi peramban pada terminal yaitu *lynx* sudah terinstal pada komputer. *Web server* akan diakses melalui jaringan *ajk* dan jaringan dalam Linux Container. Ketika berhasil diakses maka modul *web server* berhasil dilaksanakan.

### 5.2.3 Uji Kapasitas dan Performa

Uji kapasitas dan performa terdiri dari dua pengujian, yaitu uji kapasitas dan performa pada komputer *web server* dan uji

kapasitas pada komputer pembanding. Uji coba pada komputer *web server* dilakukan untuk menguji kemampuan *web server* dalam menangani simulasi ketika banyak virtualisasi berjalan pada komputer *web server* .

Sedangkan uji coba pada komputer pembanding dilakukan untuk mengetahui spesifikasi minimum komputer untuk melakukan virtualisasi menggunakan Linux Container dan seberapa banyak virtualisasi yang dapat dibuat.



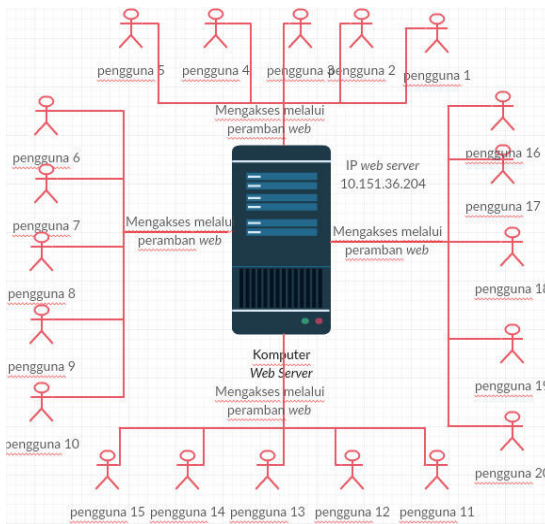
**Gambar 5.2.2 Skenario Modul Web Server**

### **5.2.3.1 Uji Kapasitas dan Performa pada Komputer Web Server**

Uji kapasitas dan performa dilakukan dengan mengakses *web server* melalui peramban *web* pada komputer klien sebanyak sepuluh. Peramban web yang digunakan dimasing- masing komputer adalah Mozilla Firefox dan Chrome. Gambar 5.2.3 merupakan desain Arsitektur uji coba kapasitas dan performa

untuk komputer *web server* . Berikut langkah – langkahnya dalam proses pengujian ini:

1. Setiap komputer melakukan *login* dengan dua akun praktikan berbeda, satu akun pada peramban Mozilla dan akun lain pada Chrome. Sehingga jumlah akun pengguna yang digunakan sebanyak 20 akun.
2. Masing – masing akun membuat sebanyak 25 virtualisasi, namun dilakukan secara bertahap yaitu, lima virtualisasi setiap tahapan.
3. Masing – masing tahap dicatat berapa persentase memori RAM yang digunakan dan persentase penggunaan prosesor dengan melihat hasil monitoring dari aplikasi htop serta waktu respon aplikasi ketika menghidupkan virtualisasi terakhir dari masing-masing tahapan. Pencatatan memori RAM dan prosesor dilakukan untuk mengetahui apakah virtualisasi dapat berjalan semestinya dan dapat diakses ketika melakukan simulasi.



**Gambar 5.2.3 Desain Arsitektur Uji Kapasitas dan Performa Komputer Web Server**

### 5.2.3.2 Uji Kapasitas dan Performa pada Komputer pembanding

Berikut langkah – langkahnya dalam proses pengujian pada komputer pembanding:

1. Skrip terdiri dari beberapa fungsi, yaitu membuat salinan dari virtualisasi yang sudah ada dan menghidupkan virtualisasi.
2. Fungsi yang dijalankan pertama kali adalah membuat salinan virtualisasi secara bertahap yaitu 10, 50, 100, 150, 200, dan 250 virtualisasi.
3. Kemudian setelah tahapan penyalinan selesai, fungsi kedua dijalankan dengan bertahap sama seperti langkah yang kedua.
4. Masing- masing tahap dicatat berapa memori RAM yang digunakan dan persentase dari prosesor yang digunakan dengan melihat hasil monitoring dari aplikasi htop.

## 5.3 Hasil Uji Coba dan Evaluasi

Pada subbab ini akan dijelaskan mengenai hasil pengujian yang dilakukan pada skenario pengujian yang telah ditentukan. Pengujian yang dilakukan adalah uji fungsionalitas dan uji kapasitas dan performa.

### 5.3.1 Hasil Uji Fungsionalitas

Tabel 5.3.1 merupakan hasil uji fungsionalitas. Uji fungsionalitas menggunakan peramban Mozilla Firefox dan fitur *add-ons* Firebug untuk melihat waktu respon yang diberikan. Tabel 5.3.2 menampilkan waktu respon rata-rata uji fungsionalitas dalam lima kali percobaan.

**Tabel 5.3.1 Hasil Eksekusi Uji Fungsionalitas**

No	Nama Kegiatan	Uji Coba	Hasil Harapan
1	<i>Login</i> sebagai	Dapat <i>login</i> ke dalam	Sukses



No	Nama Kegiatan	Uji Coba	Hasil Harapan
	admin	sistem dengan <i>username</i> dan <i>password</i> yang sudah terdaftar	
2	Login sebagai praktikan	Dapat login ke dalam sistem dengan <i>username</i> dan <i>password</i> yang sudah terdaftar	Sukses
3	Menyalin template virtualisasi	Dapat men-drag <i>template</i> dan men-drop pada kolom simulasi.	Sukses dan muncul notifikasi
4	Menghidupkan virtualisasi	Dapat menghidupkan virtualisasi	Sukses dan muncul notifikasi
5	Mematikan virtualisasi	Dapat mematikan virtualisasi	Sukses dan muncul notifikasi
6	Menghapus virtualisasi	Dapat menghapus virtualisasi	Sukses dan muncul notifikasi
7	Mengakses virtualisasi dengan SSH	Dapat mengakses virtualisasi	Sukses
8	Mengatur jaringan virtualisasi	Dapat mengatur jaringan pada komponen virtualisasi	Sukses dan muncul notifikasi
9	Mengubah konfigurasi jaringan virtualisasi	Dapat mengatur ulang jaringan pada komponen virtualisasi	Sukses dan muncul notifikasi
10	Membuat simulasi paket ICMP dengan Scapy	Dapat membuat berkas berekstensi .py yang berisi kode program Scapy	Sukses dan muncul notifikasi
11	Membuat simulasi paket TCP dengan	Dapat membuat berkas berekstensi .py yang berisi kode program	Sukses dan muncul notifikasi

No	Nama Kegiatan	Uji Coba	Hasil Harapan
	Scapy	Scapy	
12	Membuat template virtualisasi	Dapat membuat <i>template</i> virtualisasi	Sukses
13	Menambahkan praktikan	Dapat menambahkan praktikan ke dalam sistem.	Sukses
14	Menghapus praktikan	Dapat menghapus praktikan dari sistem.	Sukses
15	Menginstal aplikasi pada template virtualisasi	Dapat menginstal aplikasi pada <i>template</i> virtualisasi	Sukses
16	Menghapus template virtualisasi	Dapat menghapus template virtualisasi	Sukses
17	Mengubah pengaturan jaringan Linux Container	Dapat mengubah pengaturan jaringan Linux Container	Sukses
18	Memilih template untuk simulasi	Dapat memilih template untuk simulasi	Sukses

Tabel 5.3.2 Waktu Respon Uji Fungsionalitas

No Nama Kegiatan		Waktu Respon(ms)					
		1	2	3	4	5	Rata-rata
1	Login sebagai admin	765	881	788	1200	840	894.8
2	Login sebagai praktikan	850	774	853	807	865	829.8
3	Menyalin template	993 0	217 0	193 0	3370	725 0	4930

No Nama Kegiatan		Waktu Respon(ms)					
		1	2	3	4	5	Rata-rata
	virtualisasi						
4	Menghidupkan virtualisasi	1290	370	416	440	526	608.4
5	Mematikan virtualisasi	952	857	801	782	714	821.2
6	Menghapus virtualisasi	554	433	416	402	378	436.6
7	Mengakses virtualisasi dengan SSH	80	83	86	76	74	79.8
8	Mengatur jaringan virtualisasi	49	56	36	53	48	48.4
9	Mengubah konfigurasi jaringan virtualisasi	49	8	9	13	56	27
10	Membuat simulasi paket ICMP dengan Scapy	45	43	44	50	53	47
11	Membuat simulasi paket TCP dengan Scapy	50	47	45	42	49	46.6
12	Membuat template virtualisasi	3830	4060	7380	7720	4350	5468
13	Menambahkan praktikan	69	67	61	74	79	70
14	Menghapus praktikan	51	55	52	52	54	52.8
15	Menginstal aplikasi pada	5980	6460	5190	3940	3860	5086

No Nama Kegiatan		Waktu Respon(ms)					
		1	2	3	4	5	Rata-rata
	template virtualisasi						
16	Menghapus template virtualisasi	458	414	392	375	359	399.6
17	Mengubah pengaturan jaringan Linux Container	118	99	111	116	113	111.4
18	Memilih template untuk simulasi	123	99	112	132	112	115.6

### 5.3.2 Hasil Uji Fungsionalitas dengan Skenario Modul Praktikum

Pengujian menggunakan skenario modul praktikum jaringan komputer dilakukan oleh pengguna bernama Daniel Fablius dan Kharisma Nur Annisa. Tabel 5.3.3 merupakan hasil pengujian dengan skenario modul praktikum *web server* .

**Tabel 5.3.3 Hasil Uji Fungsionalitas dengan Skenario Modul Praktikum**

No	Nama Pengguna	NRP	Kegiatan	Keterangan dan Catatan
1	Daniel Fablius	5113100109	Menyalin komponen <i>cloud</i> , <i>switch</i> , dan komputer.	Berhasil
			Mengatur jaringan sesuai dengan modul.	Berhasil
			Melakukan ping dari <i>cloud</i> ke	Berhasil

No	Nama Pengguna	NRP	Kegiatan	Keterangan dan Catatan
			komputer.	
			Menginstal aplikasi apache2 dan php5 pada komponen <i>cloud</i> .	Berhasil
			Mengakses <i>cloud</i> melalui komputer Lab AJK dengan peramban <i>web</i> .	Berhasil
			Mengakses <i>cloud</i> melalui komponen komputer menggunakan peramban <i>web lynx</i>	Berhasil
Catatan dari pengguna : Perlu ditambahkan cara penggunaan pada halaman simulasi				
2	Kharisma Nur Annisa	5113100026	Menyalin komponen <i>cloud</i> , <i>switch</i> , dan komputer.	Berhasil
			Mengatur jaringan sesuai dengan modul.	Berhasil
			Melakukan ping dari <i>cloud</i> ke komputer.	Berhasil
			Menginstal aplikasi apache2 dan php5 pada komponen <i>cloud</i> .	Berhasil
			Mengakses <i>cloud</i> melalui komputer Lab AJK dengan	Berhasil

No	Nama Pengguna	NRP	Kegiatan	Keterangan dan Catatan
			peramban <i>web</i> .	
			Mengakses <i>cloud</i> melalui komponen komputer menggunakan peramban <i>web lynx</i> .	Berhasil
Catatan dari pengguna : tampilan koneksi dikelompokkan sesuai dengan komponen masing-masing.				

### 5.3.3 Hasil Uji Kapasitas dan Performa

Hasil uji coba kapasitas dan performa pada komputer pembanding menghasilkan maksimum virtualisasi sebanyak 100 virtualisasi dengan persentase persentase yang digunakan RAM 47,7% dan prosesor 5,1%. Hasil uji coba pada komputer *web server* jumlah virtualisasi yang berhasil dibuat melalui peramban *web* berjumlah 485, dengan 400 virtualisasi yang berjalan.

#### 5.3.3.1 Hasil Uji Kapasitas dan Performa pada Komputer Web Server

Proses uji coba berhenti ketika *web server* menampilkan *error "too many connection in MySQL database"* dengan kondisi memori pada RAM yang terpakai 84,5%. Tabel 5.3.4 menampilkan hasil uji coba kapasitas dan performa pada komputer *web server*. Waktu respon yang ditampilkan merupakan hasil pencatatan waktu respon aplikasi ketika menghidupkan virtualisasi terakhir dari masing-masing tahapan.

**Tabel 5.3.4 Hasil Uji Kapasitas dan Performa pada Komputer Web Server**

No	Jumlah Virtualisasi	Memori yang terpakai(8 GB)	Prosesor yang terpakai(4 core)	Waktu respon menghidupkan virtualisasi (ms)
1	100	22,48%	Core 1 : 8,2% Core 2 : 13,4% Core 3 : 12,2% Core 4 : 11,5%	826
2	200	36,83%	Core 1 : 22,8% Core 2 : 21,2% Core 3 : 23,2% Core 4 : 23,4%	1020
3	300	45,80%	Core 1 : 46,2% Core 2 : 44,0% Core 3 : 44,6% Core 4 : 50,8%	7510
4	400	55,13%	Core 1 : 51,7% Core 2 : 50,8% Core 3 : 50,5% Core 4 : 50,2%	12060
5	485	84,50%	Core 1 : 12,8% Core 2 : 11,9% Core 3 : 9,5% Core 4 : 13,2%	-

### 5.3.3.2 Hasil Uji Kapasitas dan Performa pada Komputer Pemanding

Tabel 5.3.5 merupakan hasil uji kapasitas dan performa pada komputer pemanding. Pada komputer pemanding, virtualisasi yang berhasil dibuat adalah 150, karena ketika hendak menjalankan virtualisasi lanjutan, tidak ada respon dari komputer pemanding ketika masukan dari *keyboard*

dijalankan sehingga proses menghidupkan virtualisasi tidak dapat berjalan lagi.

**Tabel 5.3.5 Hasil Uji Kapasitas dan Performa pada Komputer Pemanding**

No	Jumlah Virtualisasi	Memori yang terpakai(2 GB)	Prosesor yang terpakai(2 core)
1	10	9,6%	Core 1 : 0.0% Core 2 : 0.7%
2	50	37,4%	Core 1 : 3.2% Core 2 : 0.0%
3	100	47,7%	Core 1 : 0.0% Core 2 : 5.1%
4	150	61,6%	Core 1 : 12.3% Core 2 : 3.7%

#### 5.3.4 Evaluasi

Berdasarkan hasil uji fungsionalitas, sistem dapat berjalan sesuai dengan diagram kasus penggunaan yang dirumuskan pada bab 3. Rata-rata waktu yang dibutuhkan untuk mengakses sekitar lima detik. Sedangkan berdasarkan hasil uji kapasitas dan performa, komputer *web server* dapat menangani virtualisasi sebanyak 400 ketika *web server* sedang digunakan oleh 20 pengguna dengan persentase memori RAM sebesar 55, 13%, prosesor rata-rata 50% dan waktu respon ketika menghidupkan virtualisasi mencapai sekitar 12 detik. Modul praktikum jaringan komputer berupa *web server* sudah berhasil dilakukan oleh pengguna bernama Daniel Fablius dan Kharisma Nur Annisa menggunakan aplikasi simulasi jaringan ini. Sedangkan minimal spesifikasi komputer yang dapat digunakan untuk menjalankan sistem adalah komputer dengan prosesor *dual core* dan memori RAM sebesar 2 GB. Dengan jumlah maksimal virtualisasi yang dapat dibuat sejumlah 100 virtualisasi, karena pada pengujian sebanyak 150 virtualisasi, komputer tidak dapat diakses lagi.



***[Halaman ini sengaja dikosongkan]***

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan lebih lanjut.

#### **6. 1. Kesimpulan**

Dalam proses pengerjaan Tugas Akhir yang melalui tahap perancangan, implementasi, serta uji coba, didapatkan kesimpulan sebagai berikut :

1. Aplikasi simulasi jaringan menggunakan Linux Container dan manipulasi paket menggunakan Scapy untuk praktikum jaringan komputer dapat diimplementasikan dengan waktu respon rata –rata sekitar lima detik.
2. Komputer *web server* dapat menangani virtualisasi sebanyak 400 ketika *web server* sedang digunakan oleh 20 pengguna dengan persentase memori RAM sebesar 55, 13%, prosesor rata-rata 50% dan waktu respon ketika menghidupkan virtualisasi mencapai sekitar 12 detik.
3. Spesifikasi komputer yang dapat digunakan untuk menjalankan sistem adalah komputer dengan prosesor *dual core* dan memori RAM sebesar 2 GB. Dengan maksimal virtualisasi sebanyak 100 dengan persentase persentase yang digunakan RAM 47,7% dan prosesor 5,1%.

#### **6. 2. Saran**

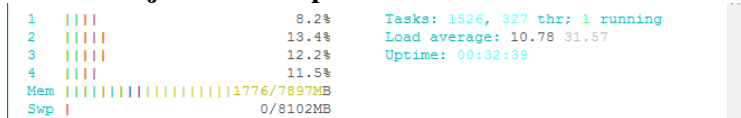
Saran yang diberikan untuk pengembangan aplikasi ini adalah :

1. Menambahkan fitur yang menampilkan simulasi jalannya pengiriman paket memanfaatkan scapy sebagai aplikasi untuk memanipulasi paket yang akan dikirimkan.

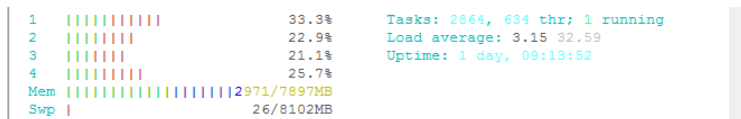
## LAMPIRAN

Bagian ini merupakan lampiran sebagai dokumen pelengkap dari buku Tugas Akhir dimana akan diberikan hasil tangkapan layar pada waktu uji coba kapasitas dan performa yang telah ditulis pada bab 5.

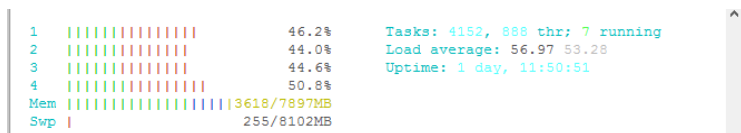
### A. Hasil Uji Coba Komputer *Web Server*



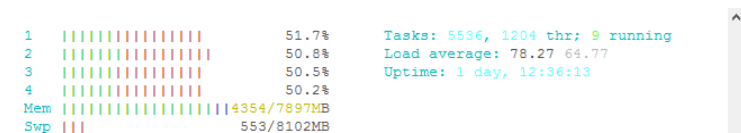
**Gambar A.1** Tangkapan Layar Hasil Uji Coba Komputer *Web Server* dengan 100 Virtualisasi



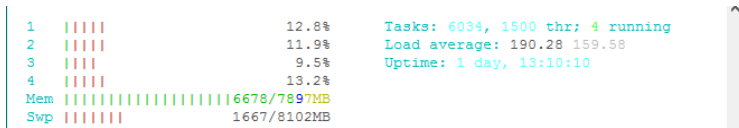
**Gambar A.2** Tangkapan Layar Hasil Uji Coba Komputer *Web Server* dengan 200 Virtualisasi



**Gambar A.3** Tangkapan Layar Hasil Uji Coba Komputer *Web Server* dengan 300 Virtualisasi

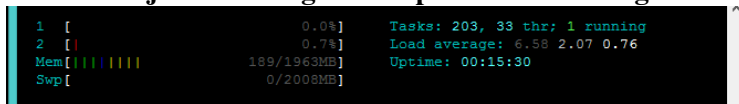


**Gambar A.4** Tangkapan Layar Hasil Uji Coba Komputer *Web Server* dengan 400 Virtualisasi

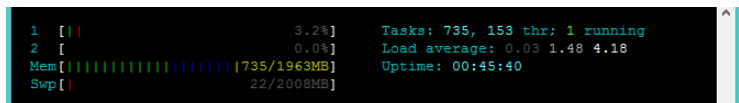


**Gambar A.5 Tangkapan Layar Hasil Uji Coba Komputer Web Server dengan 485 Virtualisasi**

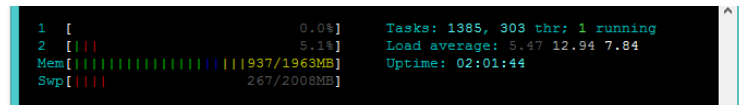
## B. Hasil Uji Coba dengan Komputer Pembanding



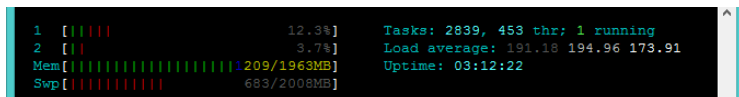
**Gambar B.1 Tangkapan Layar Hasil Uji Coba Pembanding dengan 10 Virtualisasi**



**Gambar B.2 Tangkapan Layar Hasil Uji Coba Pembanding dengan 50 Virtualisasi**



**Gambar B.3 Tangkapan Layar Hasil Uji Coba Pembanding dengan 100 Virtualisasi**



**Gambar B.4 Tangkapan Layar Hasil Uji Coba Pembanding dengan 150 Virtualisasi**

## DAFTAR PUSTAKA

- [1] M. J. Scheepers, "Virtualization and Containerization of Application Infrastructure: A Comparison," University of Twente, Enschede, 2014.
- [2] Scapy, "secdev," [secdev.org](http://www.secdev.org/projects/Scapy/), [Online]. Available: <http://www.secdev.org/projects/Scapy/>. [Diakses 6 April 2015].
- [3] J. Pan, "A Survey of Network Simulation Tools: Current Status and Future Developments," <http://www.cse.wustl.edu/>, Saint Louis, 2008.
- [4] [Linuxcontainers.org](https://Linuxcontainers.org/lxc/introduction/), "Linuxcontainers.org," [Online]. Available: <https://Linuxcontainers.org/lxc/introduction/>. [Diakses 6 April 2015].
- [5] [sourceforge.net](http://lxc.sourceforge.net/man/lxc.HTML), "sourceforge.net," [sourceforge.net](http://lxc.sourceforge.net/man/lxc.HTML), [Online]. Available: <http://lxc.sourceforge.net/man/lxc.HTML>. [Diakses 6 April 2015].
- [6] [pocoo.org](http://flask.pocoo.org/), "flask.pocoo.org," [pocoo.org](http://flask.pocoo.org/), [Online]. Available: <http://flask.pocoo.org/>. [Diakses 15 Oktober 2015].
- [7] Northwoods Software, "gojs.ne," Northwoods Software, [Online]. Available: <http://gojs.net/latest/index.HTML>. [Diakses 20 Oktober 2015].
- [8] [Linuxfoundation.org](http://www.Linuxfoundation.org/collaborate/workgroups/networking/bridge), "Linuxfoundation.org," [Online]. Available: [Linuxfoundation.org](http://www.Linuxfoundation.org/collaborate/workgroups/networking/bridge), November 2009. [Online]. Available: <http://www.Linuxfoundation.org/collaborate/workgroups/networking/bridge>. [Diakses 6 April 2015].

## BIODATA PENULIS



Dhimas Bagus Pramudya dilahirkan di Rembang 23 tahun silam. Sampai buku ini selesai ditulis, penulis masih berstatus sebagai mahasiswa Teknik Informatika angkatan 2011, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya.

Penulis memiliki pengalaman organisasi sebagai Staf Departemen Kewirausahaan dan Minat Bakat HMTC dan Instrusction Committee Kaderisasi 2013/2014 HMTC. Selain itu penulis merupakan seorang administrator di Laboratorium Arsitektur dan Jaringan Komputer. Pernah mengikuti dan mendapatkan sertifikasi HCNA-HCDA yang diselenggarakan oleh Huawei. Selama kuliah di Teknik Informatika ITS, penulis mengambil bidang minat Arsitektur dan Jaringan Komputer(AJK). Penulis pernah menjadi asisten dosen mata kuliah Sistem Operasi dan Jaringan Komputer. Penulis dapat dihubungi melalui surel di [baguspramudya@live.com](mailto:baguspramudya@live.com)